

# TEST DÉSYNCHRONISÉ SUR PROD POST 1

## Table des matières

---

---

TEST HORS SYNCHRONISATION SUR PROD POST 1  
TEST HORS SYNCHRONISATION SUR PROD POST 1  
TEST HORS SYNCHRONISATION SUR PROD POST 1  
TEST HORS SYNCHRONISATION SUR PROD POST 1  
TEST HORS SYNCHRONISATION SUR PROD POST 1

## Types basés sur l'objectif du test

### Tests fonctionnels

[Les tests fonctionnels](#) vérifient le fonctionnement du système en s'assurant que les fonctionnalités fonctionnent conformément aux exigences spécifiées.

- [Essais de fumée](#) : Effectue des contrôles de base sur les fonctionnalités critiques afin de s'assurer que l'application est stable pour des tests ultérieurs.
- [Tests d'intégrité](#) : Vérifie les fonctionnalités spécifiques après des modifications mineures ou des corrections de bogues pour s'assurer qu'elles fonctionnent comme prévu.
- [Tests d'intégration](#) : Teste l'interaction entre les différents modules pour s'assurer qu'ils fonctionnent correctement ensemble.
- [Test du système](#) : Teste l'application dans son ensemble pour valider la fonctionnalité de bout en bout.
- [Tests de régression](#) : Teste à nouveau les fonctionnalités existantes après les modifications ou les corrections de bogues pour s'assurer qu'aucun nouveau défaut n'est introduit.
- [Test d'acceptation utilisateur \(UAT\)](#) : Effectué par des utilisateurs finaux ou des clients pour vérifier que le logiciel répond aux exigences de l'entreprise.

### Essais Non Fonctionnels

[Les tests non fonctionnels](#) évaluent les performances du système en termes de convivialité, de performances, de sécurité et de compatibilité.

- [Tests de convivialité](#) : Évalue la facilité d'utilisation, l'interface utilisateur et l'expérience utilisateur globale de l'application.
- [Tests de compatibilité](#) : Vérifie que l'application fonctionne correctement sur différents navigateurs, périphériques et systèmes d'exploitation.

- [Test des performances](#) : Mesure la vitesse, la réactivité, la stabilité et le comportement de l'application sous différentes charges de travail.
- [Tests de sécurité](#) : Identifie les vulnérabilités et assure la protection des données sensibles et des ressources système.

## Processus de test manuel

Le schéma suivant fournit une représentation détaillée des étapes du processus de test manuel.

### Processus de test manuel

- Analyse des besoins : comprendre et analyser les besoins du projet, les attentes des utilisateurs et les besoins de l'entreprise pour identifier ce qui doit être testé.
- Création d'un plan de test : créez un plan de test qui définit la portée, les objectifs, les ressources, le calendrier et l'approche du test.
- Création de cas de test : conception de cas de test détaillés avec étapes de test, résultats attendus et données de test pour valider la fonctionnalité de l'application.
- Configuration de l'environnement de test : préparez le matériel, les logiciels, les outils et les données de test requis pour simuler l'environnement de production.
- Exécution des scénarios de test : exécutez les scénarios de test et comparez les résultats réels aux résultats attendus pour identifier les problèmes éventuels.
- Enregistrement des défauts : Enregistrez les défauts avec des détails complets tels que la description, la gravité, les étapes de reproduction et les preuves à l'appui.
- Correction et revérification des défauts : vérifiez que les défauts signalés ont été corrigés correctement et que le problème n'existe plus.
- Test de régression : testez de nouveau les fonctionnalités affectées et associées pour vous assurer que les modifications récentes n'ont pas introduit de nouveaux défauts.
- Rapport de clôture des tests : préparez un rapport final résumant les activités de test, les résultats, l'état des défauts et les principales leçons apprises.

## Outils utilisés pour les tests manuels

- TestRail : outil de gestion des tests permettant d'organiser, d'exécuter et de suivre les cas de tests manuels avec des fonctionnalités de création de rapports détaillées.
- Xray (pour Jira) : outil de test intégré à Jira qui prend en charge la gestion manuelle des tests, la traçabilité et le suivi des défauts.
- Qase : plate-forme de gestion des tests basée sur le cloud qui simplifie la création, l'exécution et la collaboration en équipe des scénarios de test.
- Zephyr : une solution de gestion des tests qui aide à gérer les activités de tests manuels avec une intégration Jira et des rapports solides.

- Tuskr : un outil léger de gestion des tests basé sur le cloud pour créer, gérer et suivre efficacement les cas de tests manuels.

À propos de cette traduction

Cisco a traduit ce document en traduction automatisée vérifiée par une personne dans le cadre d'un service mondial permettant à nos utilisateurs d'obtenir le contenu d'assistance dans leur propre langue.

Il convient cependant de noter que même la meilleure traduction automatisée ne sera pas aussi précise que celle fournie par un traducteur professionnel.